

INTER-DISTRIBUTED NODE LOAD DISTRIBUTION SYSTEM

Publication number: JP2000137692

Publication date: 2000-05-16

Inventor: MURATA AKIFUMI

Applicant: TOKYO SHIBAURA ELECTRIC CO

Classification:

- International: G06F15/177; G06F9/50; G06F9/54; G06F15/16;
G06F9/46; (IPC1-7): G06F15/177

- European:

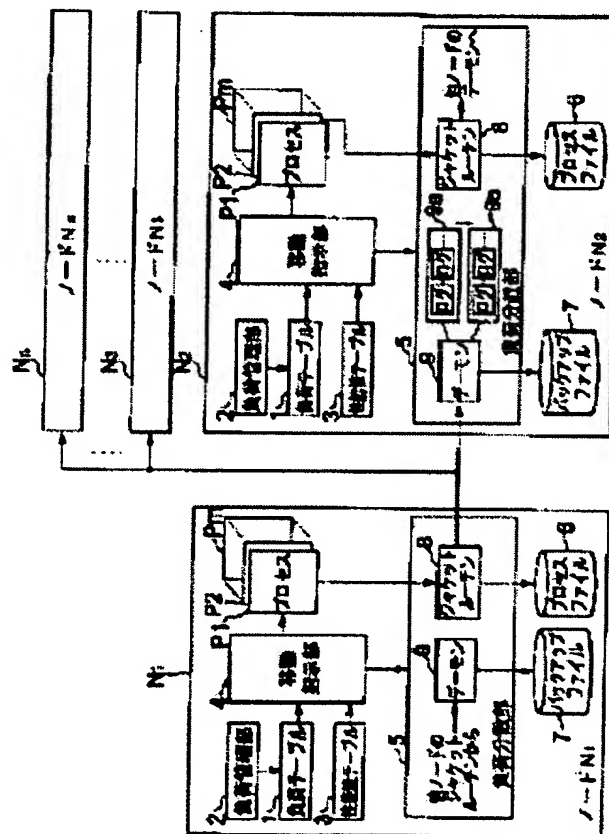
Application number: JP19980311316 19981030

Priority number(s): JP19980311316 19981030

Report a data error here

Abstract of JP2000137692

PROBLEM TO BE SOLVED: To execute the distribution of loads when loads are centralized or distribution is requested without generating any waste in a memory resource. **SOLUTION:** This inter-node load distribution system is provided with a load table 1 for storing the present load value of each node N1-Nn and a performance value table 3 for storing the processing performance value of each node N1-Nn. At the time of detecting the excess load of its own node by referring to the load table 1, a movement instructing part 4 selects a node being the destination of movement which is capable of increasing loads by referring to the performance value table 3, and generates a load movement instruction. A load distributing part 5 moves the load of its own node to the node being the destination of movement for each prescribed unit based on the load movement instruction generated by the movement instructing part 4.



Data supplied from the esp@cenet database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2000-137692

(P2000-137692A)

(43)公開日 平成12年5月16日(2000.5.16)

(51) Int.Cl.⁷

識別記号

FI

テーマコート* (参考)

G O 6 F 15/177

674

G 0 6 F 15/177

6 7 4 B

5 B 0 4 5

674C

審査請求 未請求 請求項の数 8 O L (全 10 頁)

(21)出願番号 特願平10-311316

(22) 出願日 平成10年10月30日(1998. 10. 30)

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 村田 明文

東京都府中市東芝町1番地 株式会社東芝
府中工場内

(74) 代理人 100058479

弁理士 鈴江 武彦 (外6名)

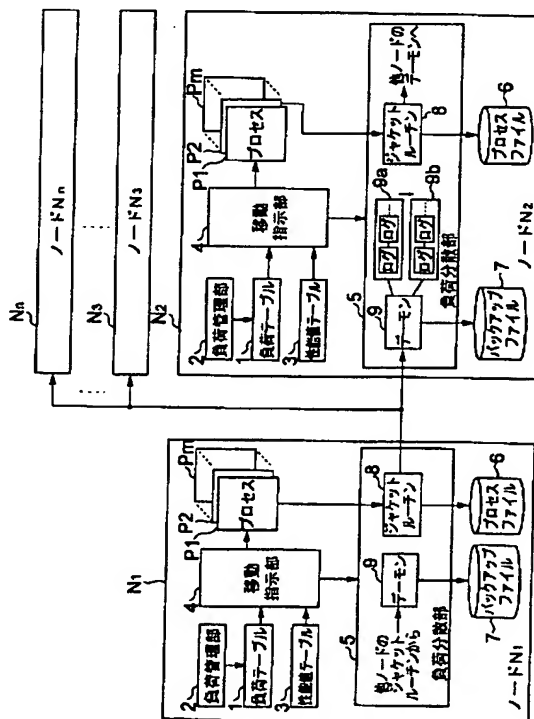
Fターム(参考) 5B045 GG04

(54)【発明の名称】 分散ノード間負荷分散方式

(57) 【要約】

【課題】 本発明は、メモリリソースに無駄を生じさせず、負荷集中時や分散要求時に負荷分散の実行を図る。

【解決手段】 各ノード $N_1 \sim N_n$ の現在の負荷値が記憶される負荷テーブル1と、各ノード $N_1 \sim N_n$ の処理性能値が記憶される性能値テーブル3とを有し、移動指示部4が、負荷テーブル1を参照して自ノードの過負荷を検出したとき、性能値テーブル3を参照して負荷を増加可能な移動先ノードを選択し、負荷移動指示を生成し、負荷分散部5が、移動指示部4により生成された負荷移動指示に基づいて、自ノードの負荷を所定単位毎に移動先ノードに移動させる分散ノード間負荷分散方式。



【特許請求の範囲】

【請求項1】 分散配置された複数のノードが互いに接続され、前記各ノードの有する負荷を各ノード間で分散させるための分散ノード間負荷分散方式であって、前記各ノードの現在の負荷値が記憶される負荷記憶手段と、前記各ノードの処理性能値が記憶される性能値記憶手段と、前記負荷記憶手段を参照して自ノードの過負荷を検出したとき、前記性能値記憶手段を参照して負荷を増加可能な移動先ノードを選択し、負荷移動指示を生成する移動指示手段と、前記移動指示手段により生成された負荷移動指示に基づいて、自ノードの負荷を所定単位毎に前記移動先ノードに移動させる負荷分散手段とを備えたことを特徴とする分散ノード間負荷分散方式。

【請求項2】 請求項1に記載の分散ノード間負荷分散方式において、前記移動指示手段は、予め複数の度合のいずれかに負荷が分類され、前記各度合毎に、移動対象の負荷が先頭にあり、移動された負荷が末尾に接続されるキューを備えたことを特徴とする分散ノード間負荷分散方式。

【請求項3】 請求項1に記載の分散ノード間負荷分散方式において、前記負荷テーブルは、前記各負荷値と、前記各負荷値における時系列的な平均値と、前記各負荷値の平均2乗誤差とが記憶されており、前記移動指示手段は、前記自ノードの過負荷を検出したとき、前記負荷テーブル内の平均値及び平均2乗誤差に基づいて、現在の負荷値と前記平均値とが所定値以上離れており、且つ前記平均2乗誤差の小さい負荷を移動対象に選択しないことを特徴とする分散ノード間負荷分散方式。

【請求項4】 請求項1に記載の分散ノード間負荷分散方式において、前記移動指示手段は、移動先対象のノードが先頭にあり、移動先にされたノードが末尾に接続されるキューを備えたことを特徴とする分散ノード間負荷分散方式。

【請求項5】 請求項1に記載の分散ノード間負荷分散方式において、前記負荷テーブルは、前記負荷値がプロセス毎に記憶され、且つ1個以上のプロセスからなる集合が前記所定単位として登録されていることを特徴とする分散ノード間負荷分散方式。

【請求項6】 請求項1に記載の分散ノード間負荷分散方式において、前記性能値テーブルは、MIPS（100万命令/秒）に基づいた前記処理性能値が記憶されることを特徴とする分散ノード間負荷分散方式。

【請求項7】 請求項1に記載の分散ノード間負荷分散

方式において、

前記移動指示手段は、自ノードにおける移動対象の負荷の負荷値、前記移動先ノードの処理性能値及び前記自ノードの処理性能値に基づいて、前記移動対象の負荷を移動した場合に前記移動先ノードで増加する負荷値を算出し、前記増加する負荷値が前記移動先ノードでのCPUのアイドル量よりも小のとき、前記負荷移動指示を生成することを特徴とする分散ノード間負荷分散方式。

【請求項8】 請求項1に記載の分散ノード間負荷分散方式において、

前記移動指示手段は、予め高負荷、中負荷又は小負荷のいずれかの度合に負荷が分類され、前記各度合毎に、移動対象の負荷が先頭にあり、移動された負荷が末尾に接続されるキューを備えたことを特徴とする分散ノード間負荷分散方式。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、分散ノードコンピューティング環境における分散ノード間負荷分散方式に関する。

【0002】

【従来の技術】一般に、複数のノードが分散配置された分散ノード間では、複数のプロセス（サービス）を実行しつつ各ノードの負荷分散を行なう際の分散ノード間負荷分散方式が知られている。

【0003】この種の分散ノード間負荷分散方式としては、例えば、RPC（リモートプロシージャコール）によるRPCサーバの分散方式がある。このRPCサーバの分散方式は、予め負荷分散用に全サーバにRPCサーバを立上げておき、全サーバに負荷を分散する方式である。また、分散ノード間負荷分散方式には、予めプロセス立上げ時に各サーバに負荷を分散する方式もある。

【0004】

【発明が解決しようとする課題】しかしながら以上のような分散ノード間負荷分散方式では、例えばRPCサーバの分散方式の場合、予め負荷分散用に全サーバにRPCサーバを立上げる必要があるため、メモリリソースに無駄を生じさせる問題があり、また、負荷集中時又は分散要求時に負荷を分散し得ない問題がある。

【0005】一方、プロセス立上げ時の分散方式の場合、負荷の集中したサーバに新たな負荷をかけないものの、この負荷の集中したサーバの負荷を分散し得ない問題がある。

【0006】本発明は上記実情を考慮してなされたもので、メモリリソースに無駄を生じさせず、負荷集中時や分散要求時に負荷分散を実行し得る分散ノード間負荷分散方式を提供することを目的とする。

【0007】

【課題を解決するための手段】請求項1に対応する発明は、分散配置された複数のノードが互いに接続され、前

記各ノードの有する負荷を各ノード間で分散させるための分散ノード間負荷分散方式であって、前記各ノードの現在の負荷値が記憶される負荷記憶手段と、前記各ノードの処理性能値が記憶される性能値記憶手段と、前記負荷記憶手段を参照して自ノードの過負荷を検出したとき、前記性能値記憶手段を参照して負荷を増加可能な移動先ノードを選択し、負荷移動指示を生成する移動指示手段と、前記移動指示手段により生成された負荷移動指示に基づいて、自ノードの負荷を所定単位毎に前記移動先ノードに移動させる負荷分散手段とを備えた分散ノード間負荷分散方式である。

【0008】また、請求項2に対応する発明は、請求項1に対応する分散ノード間負荷分散方式において、前記移動指示手段としては、予め複数の度合のいずれかに負荷が分類され、前記各度合毎に、移動対象の負荷が先頭にあり、移動された負荷が末尾に接続されるキューを備えた分散ノード間負荷分散方式である。

【0009】さらに、請求項3に対応する発明は、請求項1に対応する分散ノード間負荷分散方式において、前記負荷テーブルとしては、前記各負荷値と、前記各負荷値における時系列的な平均値と、前記各負荷値の平均2乗誤差とが記憶されており、前記移動指示手段としては、前記自ノードの過負荷を検出したとき、前記負荷テーブル内の平均値及び平均2乗誤差に基づいて、現在の負荷値と前記平均値とが所定値以上離れており、且つ前記平均2乗誤差の小さい負荷を移動対象に選択しない分散ノード間負荷分散方式である。

【0010】また、請求項4に対応する発明は、請求項1に対応する分散ノード間負荷分散方式において、前記移動指示手段としては、移動先対象のノードが先頭にあり、移動先にされたノードが末尾に接続されるキューを備えた分散ノード間負荷分散方式である。

【0011】さらに、請求項5に対応する発明は、請求項1に対応する分散ノード間負荷分散方式において、前記負荷テーブルとしては、前記負荷値がプロセス毎に記憶され、且つ1個以上のプロセスからなる集合が前記所定単位として登録されている分散ノード間負荷分散方式である。

【0012】また、請求項6に対応する発明は、請求項1に対応する分散ノード間負荷分散方式において、前記性能値テーブルとしては、MIPS（100万命令／秒）に基づいた前記処理性能値が記憶される分散ノード間負荷分散方式である。

【0013】さらに、請求項7に対応する発明は、請求項1に対応する分散ノード間負荷分散方式において、前記移動指示手段としては、自ノードにおける移動対象の負荷の負荷値、前記移動先ノードの処理性能値及び前記自ノードの処理性能値に基づいて、前記移動対象の負荷を移動した場合に前記移動先ノードで増加する負荷値を算出し、前記増加する負荷値が前記移動先ノードでのC

PUのアイドル量よりも小のとき、前記負荷移動指示を生成する分散ノード間負荷分散方式である。

【0014】また、請求項8に対応する発明は、請求項1に対応する分散ノード間負荷分散方式において、前記移動指示手段としては、予め高負荷、中負荷又は小負荷のいずれかの度合に負荷が分類され、前記各度合毎に、移動対象の負荷が先頭にあり、移動された負荷が末尾に接続されるキューを備えた分散ノード間負荷分散方式である。

（作用）従って、請求項1に対応する発明は以上のような手段を講じたことにより、各ノードの現在の負荷値が記憶される負荷記憶手段と、各ノードの処理性能値が記憶される性能値記憶手段とを有し、移動指示手段が、負荷記憶手段を参照して自ノードの過負荷を検出したとき、性能値記憶手段を参照して負荷を増加可能な移動先ノードを選択し、負荷移動指示を生成し、負荷分散手段が、移動指示手段により生成された負荷移動指示に基づいて、自ノードの負荷を所定単位毎に移動先ノードに移動させるので、従来とは異なり、メモリリソースに無駄を生じさせず、負荷集中時や分散要求時に負荷分散を実行させることができる。

【0015】また、請求項2に対応する発明は、移動指示手段としては、予め複数の度合のいずれかに負荷が分類され、各度合毎に、移動対象の負荷が先頭にあり、移動された負荷が末尾に接続されるキューを備えたので、請求項1に対応する作用に加え、特定の負荷のみが順番に移動するたらい回し動作を阻止することができる。

【0016】さらに、請求項3に対応する発明は、負荷テーブルとしては、各負荷値と、各負荷値における時系列的な平均値と、各負荷値の平均2乗誤差とが記憶されており、移動指示手段としては、自ノードの過負荷を検出したとき、負荷テーブル内の平均値及び平均2乗誤差に基づいて、現在の負荷値と平均値とが所定値以上離れており、且つ平均2乗誤差の小さい負荷を移動対象に選択しない。

【0017】これにより、請求項1に対応する作用に加え、通常は低負荷値で一時的に高負荷値となる負荷の移動を阻止できるので、移動によるオーバヘッドを抑制することができる。

【0018】また、請求項4に対応する発明は、移動指示手段としては、移動先対象のノードが先頭にあり、移動先にされたノードが末尾に接続されるキューを備えたので、請求項1に対応する作用に加え、負荷の移動先を選択する際に、他の全ノードのうち、一部のノードを検索すればよいので、サービス移動のオーバヘッドを抑制することができる。

【0019】さらに、請求項5に対応する発明は、負荷テーブルとしては、負荷値がプロセス毎に記憶され、且つ1個以上のプロセスからなる集合が所定単位として登録されているので、請求項1に対応する作用を容易且つ

確実に奏することができる。

【0020】また、請求項6に対応する発明は、性能値テーブルとしては、MIPS（100万命令/秒）に基づいた処理性能値が記憶されるので、請求項1に対応する作用を容易且つ確実に奏することができる。

【0021】さらに、請求項7に対応する発明は、移動指示手段としては、自ノードにおける移動対象の負荷の負荷値、移動先ノードの処理性能値及び自ノードの処理性能値に基づいて、移動対象の負荷を移動した場合に移動先ノードで増加する負荷値を算出し、増加する負荷値が移動先ノードでのCPUのアイドル量よりも小的时候、負荷移動指示を生成するので、請求項1に対応する作用を容易且つ確実に奏することができる。

【0022】また、請求項8に対応する発明は、移動指示手段としては、予め高負荷、中負荷又は小負荷のいずれかの度合に負荷が分類され、各度合毎に、移動対象の負荷が先頭にあり、移動された負荷が末尾に接続されるキューを備えたので、請求項1に対応する作用に加え、特定の負荷のみが順番に移動するたらい回し動作を阻止することができる。

【0023】

【発明の実施の形態】以下、本発明の各実施形態について図面を参照しながら説明する。

（第1の実施形態）図1は本発明の第1の実施形態に係る分散ノード間負荷分散方式の適用された計算機システムの構成を示す模式図である。この計算機システムは、複数のノード（計算機本体）N1～Nnが互いに接続されている。ここで、各ノードN1～Nnは、実行中のプロセス（プログラム）が異なるものの、互いに同一構成のため、ノードN1を例に挙げて説明する。

【0024】ノードN1は、実行の有無によらずに保持する複数のプロセスP1～Pmの他、負荷テーブル1、負荷管理部2、性能値テーブル3、移動指示部4、負荷分散部5、プロセスファイル6及びバックアップファイル7を備えている。

【0025】負荷テーブル1は、図2に示すように、負荷管理部2によって、各ノードN1～NnにおけるプロセスP1～Pm毎の現在のCPU負荷値、メモリ負荷値及びディスク負荷値が読出/書込可能に記憶されるテーブルであり、1以上の任意のプロセスPをまとめた集合が、1つのサービスS（負荷分散の単位）として取扱われる。

【0026】なお、負荷テーブル1は、以上の値に加え、サービスS1～Sj毎の負荷値（各サービス内の各プロセス負荷の合計値）が記憶されてもよく、ノードN1～Nn毎の負荷値（ノード内の各サービス負荷の合計値）が記憶されてもよい。

【0027】また、ノードN1全体のメモリ負荷値は、ノードN1のスワップ（swap）の使用量（スワップの残り容量あるいは一定時間のスワップin/out、ページin/out

の量）で規定可能である。ノードN1全体のCPU負荷値は、CPUのアイドル量あるいはCPU割当て待ちプロセス数で規定可能である。

【0028】負荷管理部2は、定期的に各ノードN1～NnにおけるプロセスP1～Pm毎の現在のCPU負荷値、メモリ負荷値及びディスク負荷値を収集し、これらCPU負荷値、メモリ負荷値及びディスク負荷値を負荷テーブル1に書込む機能をもっている。

【0029】性能値テーブル3は、ノードN1からNn毎に予めCPU性能値及びメモリ性能値が読出可能に登録されたテーブルであり、ノードN1～Nn内の共有領域（共有メモリでもファイルでも可）に設けられている。ここで、CPU性能値としては、 $1 / (\text{MIPS} * \text{MPU})$ が使用可能となっている。なお、MIPS（100万命令/秒）は、1つのCPUの性能値であり、MPUは、CPUの個数である。一方、メモリ性能値としては、ノード内の各メモリ容量の合計値が使用可能となっている。

【0030】移動指示部4は、定期的に負荷テーブル1を参照して自ノードN1の負荷状況を調査し、負荷が所定値を越えた旨（過負荷）を検出すると、自ノードN1の各プロセスPをサービスS単位で低負荷のノードNi（iは1～nまでの任意の自然数（但し、自ノードの番号を除く））に移動させる旨の指示を負荷分散部5に与える機能をもっている。

【0031】同様に、移動指示部4は、ノードN自体又はプロセスPの障害発生時に、障害発生により実行不可能となったプロセスPを含むサービスSを低負荷のノードNiに移動させる旨の指示を負荷分散部5に与える機能をもっている。

【0032】負荷分散部5は、移動指示部4から受けた負荷移動指示に基づいて、自ノードN1の移動対象の負荷（プロセス）をサービスS単位で他のノードNiに移動させる移動機能をもっている。

【0033】ここで、移動機能は、自ノードN1に移動対象のプロセスPk（kは1～mまでの任意の自然数）があるとき、自ノードN1における移動対象のプロセスPkを停止させ（障害により既に停止していれば不要）、移動対象のプロセスPkを立上げるための再開実行指示を移動先の他ノードNiに与え、他ノードNiにおける移動対象のプロセスPkを再開実行させることにより、結果としてプロセスPkを自ノードN1から他ノードNiに移動させるものである。

【0034】この移動機能として、負荷分散部5は、移動指示部4から受ける指示により、実行途中のプロセスPをサービスS単位で他ノードNiへ移送して低負荷の他ノードNiで継続実行する技術（特願平9-232930号）を用いている。

【0035】係る技術を用いる負荷分散部5は、プロセス実行により更新されるプロセスファイル6の更新内容

の記録（以下、ログという）を採取して他の全ノードN2～Nnに分散するためのジャケットルーチン8と、ジャケットルーチン8から受けたログを未確定キュー9aとして保持すると共に、チェックポイント毎に未確定キュー9aを確定キュー9bとして該確定キュー9b内の各ログに基づいてプロセスのバックアップファイル7を更新可能なデーモン9とを備えている。

【0036】ここで、ログは、プロセス状態を示すものであり、例えば、データ等のレジスタ情報及びファイルの更新等のシステムコール発行結果が使用可能である。チェックポイントとしては、一定時間が経過した時点、あるいはOSのシステムコール発行コールをフェッチした時点が使用可能である。

【0037】次に、以上のように構成された計算機システムの動作を説明する。ノードN1における負荷分散部5のジャケットルーチン8は、図3に示すように、実行中のプロセスP1から各ログを採取し、これら各ログをプロセスファイルに更新記憶させると共に少なくともチェックポイントCPまでに他の各ノードN2～Nnに送信する。各ノードN2～Nnでは、デーモン9がこのログを受けて未確定キューとして保持し、チェックポイントCP毎にログ内のシステムコール発行結果を反映させて処理を実行する。例えばシステムコール発行結果がバックアップファイル7の更新を示すとき、デーモン9によりバックアップファイル7を更新する。

【0038】一方、ノードN1では、移動指示部4が、定期的に負荷テーブル1の情報を監視し、一定条件（自ノードN1の負荷が所定値を越えた時点）を満たすと、負荷分散を開始する。

【0039】すなわち、移動指示部4は、図4に示すように、一定時間スリープ(sleep)し(ST1)、しかる後、負荷テーブル1を参照して自ノードN1の負荷状況を調査する(ST2)。

【0040】この調査において、自ノードN1について

$$\text{先CPUidle} > \text{前CPUload} \times \text{先CPUperf} / \text{前CPUperf} \cdots (2)$$

なお、CPUアイドル量は、新たに使用可能なCPU負荷値を意味している。すなわち、(2)式は、右辺の移動前のCPU負荷から換算される移動先のCPU負荷よりも、左辺の移動先のCPUアイドル量が大である関係を意味している。

【0047】また、サービス移動後の移動先ノードNi

$$\text{後CPUidle} + 100 \times (\text{Mtotal} - \text{後Muse}) / \text{Mtotal} \cdots (3)$$

すなわち、(3)式は、サービス移動後において、CPU負荷の余裕分と、メモリ負荷の余裕分とを合計した値を示している。

【0049】ステップST6においては、(1)式、(2)式を共に満たした場合、すなわち、メモリ負荷及びCPU負荷を共に移動可能と判定したときのみ、サービスSを移動可能と判定し、負荷分散部5にサービス移動の指示を出し(ST7)、ステップST1へ戻る。

過負荷か否かを判定し(ST3)、過負荷でないときにはステップST1へ戻る。

【0041】なお、ステップST3の判定は、メモリ負荷の場合、前述したノードN1全体のメモリ負荷値が所定値を越えたときに過負荷とし、CPU負荷の場合、前述したノードN1全体のCPU負荷値が所定値を越えたときに過負荷とする。

【0042】自ノードN1を過負荷と判定したとき、自ノードN1から高負荷のサービスSを他ノードNiへの転送対象として選択する(ST4)。

【0043】続いて、最低の負荷の例えばノードN2を選択し(ST5)、負荷テーブル1及び性能値テーブル3を参照しつつ、サービスSを移動可能か否かを判定する(ST6)。

【0044】ステップST6の判定は、メモリ負荷と、CPU負荷との2通りが実行される。メモリ負荷の判定は、移動先ノードN2のメモリ性能値をMtotalとし、移動先ノードN2で現在使用中のメモリ負荷値をMusin gとし、移動するサービスSのメモリ負荷値をMservとした場合、次の(1)式を満たすときに移動可能とされる。

$$\text{Mtotal} - \text{Musin g} > \text{Mserv} \cdots (1)$$

なお、メモリ性能値はノードN2内の各メモリ容量の合計値である。

【0045】次に、CPU負荷の判定は、移動先ノードN2の現在のCPUアイドル(遊休)量を先CPUidleとし、移動前のノードN1でのサービスSのCPU負荷値を前CPUloadとし、移動先ノードN2のCPU性能値を先CPUperfとし、移動前のノードN1でのCPU性能値を前CPUperfとした場合、次の(2)式を満たすときに移動可能とされる。

【0046】

【数1】

の評価において、CPUアイドル量を後CPUidleとし、メモリ負荷値を後Museとした場合、各ノードN2～Nnのうち、次の(3)式の値が最高のノードN2が、最低の負荷のノードN2として判定される。

【0048】

【数2】

【0050】なお、ステップ6において、高負荷のサービスSを移動できないとき、中程度の負荷のサービスSを選択し(ST8)、前述同様にサービスSを移動可能か否かを判定する(ST9)。

【0051】また、ステップST9において移動可能ときにはステップST7に行くが、中程度の負荷のサービスSが移動不可のとき、低負荷のサービスSを選択し(ST10)、前述同様にサービスSを移動可能か否かを判定する(ST9)。

を判定する(ST11)。

【0052】ステップST11においても、移動可能なときにはステップST7に行くが、低負荷のサービスSが移動不可のとき、ステップST1へ戻る。

(具体例1) 次に、以上のような各ステップST1～ST11において、1つのプロセスP1のみを有する1つのサービスS1の移動に際し、CPU負荷のみを検討する場合について説明する。

【0053】具体的には、図5に示す負荷状況において、ノードN1のCPU負荷値が90%を越えた際に、サービスS1を移動させる場合の移動指示部4の動作を述べる。

【0054】ステップST3において、ノードN1のCPU負荷は、サービスS1～S3を足して90%であるため、ノードN1が過負荷と判定される。また、ステップST4において、ノードN1で最も高負荷のサービスS1が転送対象として選択される。

【0055】次いで、ステップST5において、最低の負荷のノードN2が選択される。例えば、サービスS1を他ノードN2又はN3へ移動した場合を仮定し、ノードN2、N3にてサービスS1を実行する場合のCPU負荷値を試算する。その試算結果は、ノードN2が25% ($= 50\% * (1/200) / (1/100)$) であり、ノードN3が16% ($= 50\% * (1/300) / (1/100)$) である。

【0056】ここで、サービスS1を移動すると、最終的なCPU負荷値は、ノードN2では35% ($= 10\% + 25\%$) となり、ノードN3では46% ($= 30\% + 16\%$) となる。従って、最終的なCPU負荷値の小さいノードN2は、ステップST5により最低の負荷のノードN2として選択され、ステップST6の(2)式により ($\text{先CPU負荷} 90\% > 25\%$) 移動可能と判定され、ステップST7によりサービスS1が移動される。

(具体例2) また、具体例と同一のCPU性能値において、他のサービスが移動される場合について説明する。図6に示す負荷状況において、サービスS1の移動を仮定した場合、各ノードN2、N3の負荷は、ノードN2が110% ($= 85\% + 25\%$) となり、ノードN3が91% ($= 75\% + 16\%$) となる。この場合、ノードN2、N3の負荷が高いので、サービスS1を移動できない。

【0057】よって、中程度の負荷であるサービスS2の移動を検討する。サービスS2を移動した場合の計算は、ノードN2が100% ($= 85\% + 15\%$) となり、ノードN3が85% ($= 75\% + 10\%$) となるので、サービスS2をノードCへ移動させる。

【0058】このように負荷が高い順から、サービスS1、…の移動を計算し、移動可能なノードN3にサービスを移動させる。但し、全てのサービスS1～S3が移動不可能(他の全ノードN2、N3が高負荷状態)のとき、サービスS1～S3の移動をあきらめる。

き、サービスS1～S3の移動をあきらめる。

【0059】このように、ノードN1では、ノードN1自体又はプロセスPkにて障害発生あるいは高負荷の発生により、プロセスPkの実行が困難になると、(高負荷の発生時には予め当該プロセスPkを停止させた後、) 低負荷の例えばノードN2にプロセス移動を指示して負荷分散を実行する。ノードN2では、プロセスの再開を実行する。

【0060】ノードN2は、再開の実行時に、プロセスPkのmain(プログラムとしてのスタート)をフェッチし、チェックポイントCPのログからスタック積上げ/レジスタ情報設定等を実行し、ノードN1で中止されたプロセスPkを最新のチェックポイントCP時点から再開して実行する。

【0061】上述したように本実施形態によれば、各ノードN1～Nnの現在の負荷値が記憶される負荷テーブル1と、各ノードN1～Nnの処理性能値が記憶される性能値テーブル3とを有し、移動指示部4が、負荷テーブル1を参照して自ノードの過負荷を検出したとき、性能値テーブル3を参照して負荷を増加可能な移動先ノードを選択し、負荷移動指示を生成し、負荷分散部5が、移動指示部4により生成された負荷移動指示に基づいて、自ノードの負荷を所定単位毎に移動先ノードに移動させるので、従来とは異なり、メモリリソースに無駄を生じさせず、負荷集中時や分散要求時に負荷分散を実行させることができる。

【0062】また、プロセスの実行中の移動により、サービスの継続性を保ちつつ、プログラミングによる負荷分散の意識をせずに、分散ノード間での負荷分散システムを構築することができる。

(第2の実施形態) 次に、本発明の第2の実施形態に係る分散ノード間負荷分散方式の適用された計算機システムについて説明する。

【0063】本実施形態は、第1の実施形態中、各ノードN1～Nnの平均の負荷よりも高負荷のサービスSがある場合、この高負荷のサービスSのみが各ノードN1～Nnを順番に移動する(たらい回しされる)場合があることを考慮し、このたらい回し動作の阻止を図るものである。

【0064】具体的には、移動指示部4は、前述した機能に加え、図7に示すように、例えば各サービスS1～S9が負荷の程度に応じて配列される高、中、低の3段階のキューQ1～Q3を有し、各段階の負荷のサービスを選択(ST4, 8, 10)する際に、各段階のキューの先頭にあるサービスS7(S8又はS9)を移動対象として選択する機能と、サービスS7(S8又はS9)が移動されたときにはこのサービスS7(S8又はS9)を該当する段階のキューQ1(Q2又はQ3)の末尾に接続する機能とをもっている。

【0065】なお、図7中のサービスSの添字及び各Q

1～Q3内のサービスSの個数は、単なる一例であり、適宜変更可能なことは言うまでもない。次に、以上のように構成された計算機システムの動作を説明する。なお、この説明は、第1の実施形態と比較して述べる。

【0066】前述した第1の実施形態の場合、図5と同一のCPU性能値のノードN1において、図8に示すCPU負荷状況であるとする。この場合、ノードN1は、サービスS1をノードN2に移動させる。ここで、サービスS1のCPU負荷値が35%～45%の範囲内で上下すると、ノードN2は、サービスS1を他のノードN3に移動させる可能性がある。また、ノードN3はサービスS1をさらに他のノードN4に移動させる。以下同様に、サービスS1のみが各ノードN5～Nnを順番に移動する可能性がある。

【0067】一方、本実施形態では、サービス選択用のキューを設けた構成により、移動対象のサービスS1～S9が図9に示すようにキューQ1～Q3に接続される。

【0068】ここで、ノードN1がキューQ1の先頭のサービスS1をノードN2に移動させると、ノードN2では、図10に示すように、このサービスS1がキューQ1の最後に接続される。

【0069】これにより、ノードN2がサービスSを移動させる場合、高負荷のキューQ1の先頭であるサービスS6が移動対象となる。従って、第1の実施形態とは異なり、サービスS1のみが順番に移動するたらい回し動作を阻止することができる。

【0070】上述したように本実施形態によれば、第1の実施形態の効果に加え、あるサービス（例えば全ノード中で一番負荷の高いサービス）のみがたらい回しにされる動作を阻止することができる。

（第3の実施形態）次に、本発明の第3の実施形態に係る分散ノード間負荷分散方式の適用された計算機システムについて説明する。

【0071】本実施形態は、第1の実施形態中、通常は低負荷で一時的に高負荷になるが直ぐに低負荷に復帰するサービスSがある場合、このサービスSを移動させる場合があることを考慮し、この一時的に高負荷となるサービスSの移動の阻止を図るものである。

【0072】具体的には、負荷テーブル1は、前述した現在の負荷状況に加え、過去の負荷状況の平均値及び平均2乗誤差が記憶されるものである。

【0073】負荷管理部2は、前述した機能に加え、過去の負荷状況の平均値及び平均2乗誤差を負荷テーブル1に書込む機能をもっている。

【0074】移動指示部4は、前述した機能に加え、ステップS T4でノードが過負荷か否かを判定する際に、現在の負荷値と負荷値の平均値とが著しく離れており、且つ平均2乗誤差が小さいノードNを選択しない機能を有している。

【0075】次に、以上のように構成された計算機システムの動作を説明する。なお、この説明は、第1の実施形態と比較して述べる。

【0076】いま、図11に示すように、通常は低負荷で一瞬だけ高負荷になるサービスS1があるとする。このサービスS1は、一瞬だけ負荷が上昇したが、通常は低負荷であるので、サービスS1を移動せずに時間の経過を待つ方がよい。

【0077】しかし、第1実施形態では、高負荷となる時間Aにおいて、移動対象のサービスS1を選択する場合、このサービスS1を移動対象とする可能性がある。

【0078】一方、本実施形態では、負荷の平均値と平均2乗誤差とを管理する構成により、移動対象のサービスSkを選択する際に、現在の負荷と負荷の平均値とが著しく離れており、且つ平均2乗誤差が小さいサービスS1を移動対象に選択しない。

【0079】これにより、通常は低負荷で一時的に高負荷となるサービスS1の移動を阻止できるので、移動によるオーバヘッドを抑制することができる。

【0080】上述したように本実施形態によれば、第1の実施形態の効果に加え、通常は低負荷値で一時的に高負荷値となる負荷の移動を阻止できるので、移動によるオーバヘッドを抑制することができる。

（第4の実施形態）次に、本発明の第4の実施形態に係る分散ノード間負荷分散方式の適用された計算機システムについて説明する。

【0081】本実施形態は、第1の実施形態中、多数のノードN1～Nnを有する計算機システムの場合、最低の負荷をもつノードNiを選択する際に、ノード数nに比例してノードN1～Nnの負荷を算出する処理のオーバヘッドを増大させることを考慮し、このオーバヘッドの抑制を図るものである。

【0082】具体的には、移動指示部4は、前述した機能に加え、移動対象の例えばノードN1～N4を順番に配列したキューQmを有し、前述したステップS T5によるノード選択の際に、最低の負荷のノードN2を選択するのではなく、キューQmの先頭にあるノードNsを移動先として選択する機能と、キューQmの先頭のノードNにサービスSを移動可能ととき、サービスSをその先頭のノードNに移動させる機能と、サービスSを移動させたノードNをキューQmの末尾に接続する機能とを有している。

【0083】次に、以上のように構成された計算機システムの動作を説明する。

【0084】本実施形態では、ノード選択用のキューQを設けた構成により、移動先の候補としてノードN1～Nxが待ち行列に接続される。

【0085】例えば、ノードN1→ノードN2→ノードN3→ノードN4というキューQmがあり、ノードN2から高負荷のサービスS1をノードN1に移動するとす

る。

【0086】ここで、移動指示部4は、ノード選択の際に、キューQmの先頭にあるノードN1を移動先として選択し、そのノードN1の負荷を算出してそのノードN1にサービスSを移動可能なとき、サービスSをそのノードN1に移動させる。

【0087】また、移動指示部4は、サービスSの移動により負荷の増えたノードN1をキューQmの末尾に接続する一方、サービスSを移動して負荷の減ったノードN2をキューQmの先頭に接続する。サービス移動後のキューQmの状態は、ノードN1→ノードN3→ノードN4→ノードN2のようになる。

【0088】このように、負荷を移動したノードがキューの先頭へ接続され、負荷の移動されたノードはキューの最後に接続されることにより、低負荷のノードがキューQmの先頭へ配置され、高負荷のノードがキューQmの後半に配置される。

【0089】従って、次回、負荷移動先を算出する場合もキューQmの先頭からのヒット率が高くなり、全ノードN1～Nnの負荷を算出する場合に比べ、オーバーヘッドを抑制することができる。

【0090】上述したように本実施形態によれば、第1の実施形態の効果に加え、ノードN2がサービスSの移動先を選択する際に、他の全ノードN1、N3～Nnのうち、一部のノードを検索すればよいので、サービス移動のオーバーヘッドを抑制することができる。

【0091】なお、上記各第2～第4の実施形態は、第1の実施形態に個別に適用した場合を説明したが、これに限らず、適宜組合せて同時に適用する構成としても、本発明を同様に実施して同様の効果を得ることができる。

【0092】また、上記実施形態に記載した手法は、コンピュータに実行させることのできるプログラムとして、磁気ディスク（フロッピーディスク、ハードディスクなど）、光ディスク（CD-ROM、DVDなど）、光磁気ディスク（MO）、半導体メモリなどの記憶媒体に格納して頒布することもできる。

【0093】その他、本発明はその要旨を逸脱しない範囲で種々変形して実施できる。

【0094】

【発明の効果】以上説明したように本発明によれば、メモリリソースに無駄を生じさせず、負荷集中時や分散要

求時に負荷分散を実行できる分散ノード間負荷分散方式を提供できる。

【図面の簡単な説明】

【図1】本発明の第1の実施形態に係る分散ノード間負荷分散方式の適用された計算機システムの構成を示す模式図

【図2】同実施形態における負荷テーブルの構成を示す模式図

【図3】同実施形態における動作を説明するための模式図

【図4】同実施形態における移動指示部の動作を説明するためのフローチャート

【図5】同実施形態における動作を説明するための負荷状況を示す模式図

【図6】同実施形態における動作を説明するための負荷状況を示す模式図

【図7】本発明の第2の実施形態に係る分散ノード間負荷分散方式に用いられるキューの内容を示す模式図

【図8】同実施形態における動作を説明するための負荷状況を示す模式図

【図9】同実施形態における動作を説明するためのキューの内容を示す模式図

【図10】同実施形態における動作を説明するためのキューの内容を示す模式図

【図11】本発明の第3の実施形態に係る分散ノード間負荷分散方式を説明するためのサービスの負荷値を示す模式図

【符号の説明】

- 1…負荷テーブル
- 2…負荷管理部
- 3…性能値テーブル
- 4…移動指示部
- 5…負荷分散部
- 6…プロセスファイル
- 7…バックアップファイル
- 8…ジャケットルーチン
- 9…デーモン
- 9a…未確定キュー
- 9b…確定キュー
- N1～Nn…ノード
- P1～Pm…プロセス

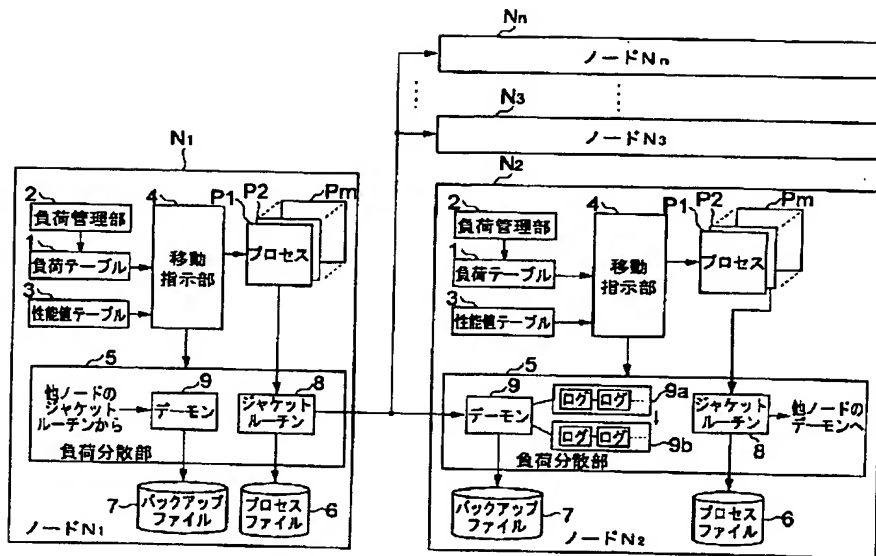
【図2】

ノード	サービス	プロセス	CPU負荷	メモリ負荷	ディスク負荷
ノードN1	サービスS1	プロセスP1			
		プロセスP2			
	サービスS2	プロセスP3			
		プロセスP4			
ノードN2	サービスS3	プロセスP5			

【図6】

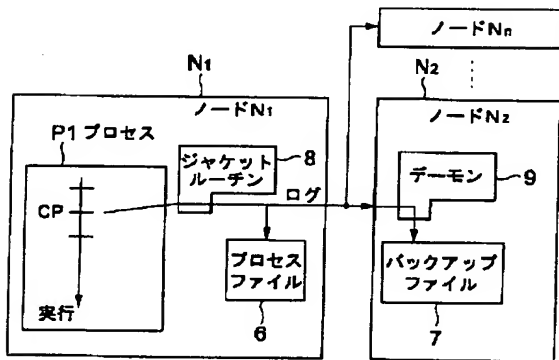
ノードN1	ノードN2	ノードN3
サービスS1 60%	サービスS4 85%	サービスS5 75%
サービスS2 30%		
サービスS3 10%		

【図1】



【図3】

【図4】

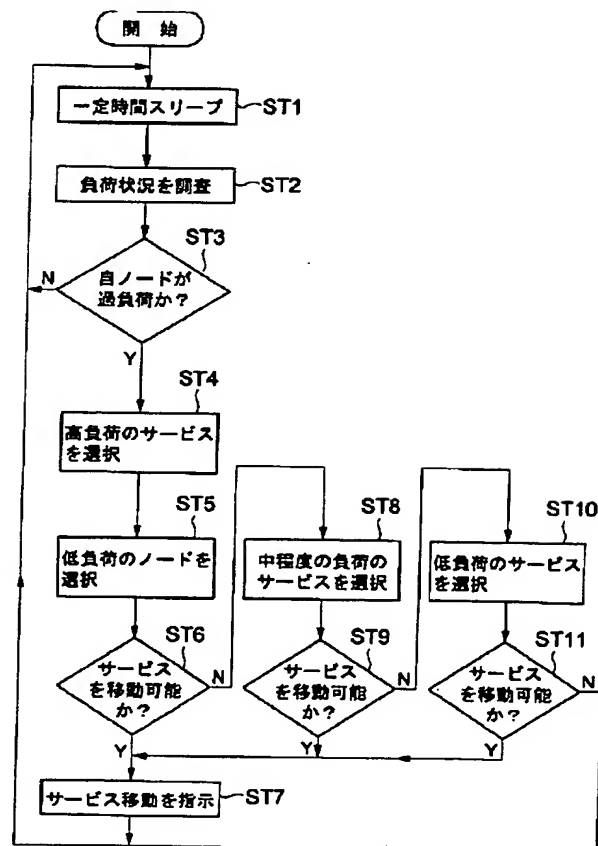


【図8】

ノードN1	ノードN2	ノードN3
サービスS1 35%	サービスS8 30%	
サービスS2 30%		
サービスS3 20%	サービスS7 20%	サービスS8 28%
サービスS4 10%		サービスS9 20%
サービスS5 5%		

【図9】

ノードN1	ノードN2	ノードN3
高負荷のキューQ1 サービスS2→サービスS1	サービスS6	
中負荷のキューQ2 サービスS3→サービスS4	サービスS7	サービスS8→サービスS9
低負荷のキューQ3 サービスS5		



【図5】

	ノードN1	ノードN2	ノードN3
1 負荷 テーブル	サービス CPU負荷	サービス CPU負荷	サービス CPU負荷
	サービスS1 50%	サービスS4 10%	サービスS5 30%
	サービスS2 30%		
	サービスS3 10%		
3 性能値 テーブル	CPU性能 (100MHz 1個)	CPU性能 (100MHz 2個)	CPU性能 (300MHz 1個)
	1/100	1/200	1/300

【図10】

ノードN1	ノードN2	ノードN3
高負荷のキューQ1 サービスS2	サービスS1→サービスS8	
中負荷のキューQ2 サービスS3→サービスS4	サービスS7	サービスS8→サービスS9
低負荷のキューQ3 サービスS5		

【図7】

高負荷のキュー Q1	サービスS1→サービスS4→サービスS7
中程度の負荷のキュー Q2	サービスS2→サービスS5→サービスS8
低負荷のキュー Q3	サービスS3→サービスS6→サービスS9

【図11】

